

CONCEPTION D'UNE SOCIÉTÉ D'AGENTS POUR UNE MÉMOIRE D'ENTREPRISE

Fabien Gandon, Rose Dieng-Kuntz
Projet ACACIA – INRIA Sophia Antipolis
{Fabien.Gandon|Rose.Dieng}@sophia.inria.fr

1 Introduction

Le projet CoMMA¹ s'intéresse au développement d'un environnement de gestion de la mémoire d'entreprise conjuguant au sein d'un système multi-agents (SMA) des techniques de l'ingénierie des connaissances, de la galaxie XML, de la recherche d'information et de l'apprentissage symbolique. Lorsqu'il envisage une solution logicielle dans une perspective multi-agents, le concepteur doit gérer la relation entre :

- le *niveau macroscopique du SMA (la société des agents)*, où se posent les problèmes d'ingénierie des interactions et d'organisation de la société du SMA afin d'obtenir, du point de vue global du système, les fonctionnalités correspondant aux exigences de l'utilisateur.
- le *niveau microscopique du SMA (les agents en eux-mêmes)* où se posent les problèmes d'identification des rôles nécessaires, d'ingénierie des comportements tenant compte des interactions qui se produiront et fournissant les différentes compétences recherchées.

Les fonctionnalités souhaitées pour le système ne se transfèrent pas directement en fonctionnalités d'agents, mais influencent la conception et sont finalement distribuées dans les interactions sociales des agents et l'ensemble des capacités, des rôles et des comportements qui leurs sont associés. Nous résumons ici l'analyse de conception menée dans CoMMA.

2 Du macroscopique au microscopique

L'*architecture* d'un système multi-agents dépeint les différentes familles d'agents possibles et les rapports qu'ils entretiennent. Une *configuration* est une instance d'une architecture avec un agencement choisi et un nombre approprié d'agents de chaque type. L'architecture doit permettre plusieurs configurations de déploiement couvrant les différentes structures organisationnelles envisageables. La description d'une configuration peut être étudiée et documentée lors du déploiement en utilisant des diagrammes de déploiement UML. L'architecture est étudiée et fixée lors de la conception. L'analyse architecturale part du plus haut niveau d'abstraction du système (i.e. la société) et par raffinements et décompositions successifs (i.e. sous-sociétés gigognes) elle descend jusqu'au point où les rôles des agents et leurs interactions peuvent être identifiés et spécifiés. En considérant les fonctionnalités du système CoMMA, nous avons identifié quatre sous-sociétés : (1) une sous-société dédiée au modèle d'entreprise et à l'ontologie, (2) une sous-société dédiée aux documents, (3) une sous-société dédiée aux utilisateurs, (4) une sous-société dédiée à l'interconnexion entre agents.

Lors de l'analyse des sous-sociétés dédiées aux ressources (modèles, documents et pages jaunes), nous avons constaté trois types récurrents d'organisations. (1) *La société hiérarchique* distingue deux types d'agents : Les représentants, médiateurs entre leur société et le reste du SMA ; Les exploitants, assignés à une ressource locale, ils répondent aux requêtes du médiateur. (2) *La société égalitaire* repose sur des relations égalitaires entre des rôles complètement redondants. N'importe quel agent peut être contacté de l'extérieur de la société pour répondre à une requête concernant le type de ressources auquel sa société est dédiée ; charge à lui de s'associer avec d'autres agents pour répondre à la requête. (3) *La société de duplication* où ni les rôles ni le contenu ne sont distribués, repose sur des agents qui maintiennent à jour une copie complète de toute l'information et peuvent résoudre les requêtes seuls. Selon les contraintes imposées, une sous-société dédiée sera organisée selon l'un ou l'autre des modèles précédents, et les interactions exigeront différents protocoles (requête, question, appel à proposition, enchères, etc.).

¹Corporate Memory Management through Agents - projet IST 1999-12217 subventionné par la Commission Européenne

Les agents de la *sous-société dédiée au modèle d'entreprise et à l'ontologie* fournissent des services de diffusion, de mise à jour et de requêtes sur l'ontologie et le modèle. Pour cette société, les trois types d'organisation sont envisageables. Dans le premier prototype de CoMMA, nous avons opté pour une société de duplication.

Les agents de la *sous-société dédiée aux documents* sont en charge de l'exploitation des bases d'annotations distribuées dans l'entreprise. Ces agents recherchent et extraient les références satisfaisant la requête de l'utilisateur. Pour cette société, seules les deux premières options sont envisageables ; la société de duplication n'est pas réaliste, car cela signifierait la duplication d'une image complète de toute la mémoire pour chaque agent de cette société. Le système CoMMA a opté pour une organisation hiérarchique.

Les agents de la *sous-société d'interconnexion* sont responsables de l'appariement des autres agents en se basant sur l'expression de leurs besoins et la description de leurs capacités. Dans JADE², la plate-forme agent utilisée dans CoMMA, les agents responsables des pages jaunes sont fédérés en une société égalitaire.

Les agents de la *sous-société dédiée aux utilisateurs* sont en charge des aspects d'interface, d'observation, d'aide et d'adaptation à l'utilisateur. N'étant pas liés à la gestion d'une ressource, ils ne suivent pas la typologie précédente des sous-sociétés. Nous nous sommes intéressés en priorité à deux rôles récurrents dans les systèmes d'information multi-agents : le rôle de gestion de l'interface utilisateur ; le rôle de gestion du profil utilisateur.

3 Des rôles et interactions aux comportements

La deuxième étape de notre approche est de dériver de l'analyse de l'architecture les caractéristiques des rôles et les protocoles de leurs interactions, afin de choisir une implantation des comportements de chaque type d'agents.

Un *rôle* représente la position d'un agent dans une société et les responsabilités et les activités, assignées à cette position, que les autres agents s'attendent à être correctement remplies. L'analyse des rôles est à la charnière entre le niveau microscopique et le niveau macroscopique. Dans les sous-sociétés, nous avons identifié les rôles suivants : (1) *Archiviste d'ontologie* : gestion et accès à l'ontologie. (2) *Archiviste Modèle d'Entreprise* : gestion et accès au modèle d'entreprise. (3) *Archiviste d'Annotations* : gestion et accès à une base d'annotations. (4) *Médiateur d'Annotations* : médiation entre les archivistes et les demandeurs. (5) *Apparieur* : maintenance et accès aux pages jaunes. (6) *Contrôleur d'Interface* : contrôle de l'interface graphique utilisateur. (7) *Gestionnaire de profils* : modification des profils d'utilisateurs connectés. (8) *Archiviste de profils* : archivage et accès aux profils utilisateurs.

Après la spécification des rôles, nous nous intéressons aux *interactions* entre agents. Les modèles des conversations sont spécifiés par des *protocoles*. La spécification d'un protocole part d'un graphe d'acointances au niveau des rôles, qui représente par un graphe orienté les voies de communication existant entre les agents jouant ces rôles. Sur ce graphe, on spécifie la séquence possible des messages échangés pendant un type d'interaction donné.

A partir de la description des rôles et des interactions, nous avons proposé et implanté des *types d'agents* remplissant un ou plusieurs rôles. Le *comportement* d'un type d'agent combine les différents comportements implantés pour remplir les activités correspondant aux rôles qui lui sont assignés. Les comportements intègrent des outils de différents domaines de recherche comme briques de base pour ajouter aux différents types d'agents les capacités techniques nécessaires pour remplir leurs rôles.

Pointeurs :

- Corporate Memory Management through Agents, *Actes de la Conférence E-Work & E-Business*, Madrid, Octobre 2000
- Gandon, Dieng, Corby, Giboin, A Multi-Agents System to Support Exploiting an XML-based Corporate Memory, *Actes 3rd International Conference on Practical Aspects of Knowledge Management*, 2000, Basel, p.10-1 to 10-12

² Java Agent DEvelopment framework - <http://sharon.cselt.it/projects/jade/>